

Mlfuns Sample

Phase 1 Data Assembly

August 10, 2010

Tim Bergsma

This script assembles simulated phase 1 data.

Make sure you are in the script directory, where this files resides.

Listing 1:

```
> getwd()

[1] "/Users/timb/project/metrum-mifuns/inst/sample/script"
```

Load the Mifuns package.

Listing 2:

```
> library(Mifuns)

Mifuns 4.0.11 loaded
Installing SIGCHLD signal handler...Done.
```

Groom the dose data

Listing 3:

```
> dose <- read.csv('../data/source/dose.csv', na.strings='.', stringsAsFactors=FALSE)
> head(dose)

  SUBJ  AMT HOUR
1    1 1e+03    0
2    2 5e+03    0
3    3 1e+04    0
4    4 5e+04    0
5    5 1e+05    0
6    6 1e+03    0
```

Listing 4:

```
> dose <- as.keyed(dose, key=c('SUBJ', 'HOUR'))
> summary(dose)

SUBJ~HOUR
0 NA keys
0 duplicate keys
```

Looks okay.

Groom the demographic data.

Listing 5:

```
> dem <- read.csv('../data/source/dem.csv', na.strings='.', stringsAsFactors=FALSE)
> head(dem)
```

```

SUBJ HEIGHT WEIGHT SEX AGE DOSE FED SMK DS CRCN
1      1    174   74.2  0 29.1 1e+03  1  0  0  83.5
2      2    177   80.3  0 36.8 5e+03  1  0  0 142.0
3      3    180   94.2  0 46.4 1e+04  1  0  0 121.0
4      4    177   85.2  0 30.3 5e+04  1  0  0 127.0
5      5    166   82.8  0 32.5 1e+05  1  0  0  97.2
6      6    164   63.9  0 18.8 1e+03  1  0  0 138.0

```

Listing 6:

```

> dem <- as.keyed(dem, key='SUBJ')
> summary(dem)

```

```

SUBJ
0 NA keys
0 duplicate keys

```

Looks okay. Note that DOSE is a treatment group, not an actual dose.

Groom the pk data.

Listing 7:

```

> pk <- read.csv('../data/source/pk.csv', na.strings='.', stringsAsFactors=FALSE)
> head(pk)

```

```

SUBJ HOUR DV
1      1 0.00 0.000
2      1 0.25 0.363
3      1 0.50 0.914
4      1 1.00 1.120
5      1 2.00 2.280
6      1 3.00 1.630

```

Listing 8:

```

> pk <- as.keyed(pk, key=c('SUBJ', 'HOUR'))
> head(pk)

```

```

SUBJ HOUR DV
1      1 0.00 0.000
2      1 0.25 0.363
3      1 0.50 0.914
4      1 1.00 1.120
5      1 2.00 2.280
6      1 3.00 1.630

```

Listing 9:

```

> summary(pk)

```

```
SUBJ~HOUR
1 NA keys
2 duplicate keys
unsorted
```

Listing 10:

```
> pk[naKeys(pk),]
```

```
      SUBJ HOUR  DV
561     40   NA 100
```

Listing 11:

```
> pk[dupKeys(pk),]
```

```
      SUBJ HOUR  DV
560     40   72 35.5
562     40   72   NA
```

Listing 12:

```
> bad <- pk[with(pk,is.na(HOUR) |is.na(DV)),]
> bad
```

```
      SUBJ HOUR  DV
561     40   NA 100
562     40   72   NA
```

Listing 13:

```
> pk <- pk - bad
```

```
dropping 2 of 562 rows matching on SUBJ, HOUR, DV
```

Listing 14:

```
> summary(pk)
```

```
SUBJ~HOUR
0 NA keys
0 duplicate keys
```

Looks okay.

Combine these data sources into an NMTRAN-style data set. The function 'aug' adds columns on-the-fly. The function 'as.nm' sets up a chain reaction that makes sure the final result has properties of an NMTRAN data set as described in ?nm.

Every source must specify DATETIME or HOUR. All of ours specify HOUR. If HOUR is the same for two records, we want, e.g., pk samples to sort before dose records (assumed predose). SEQ controls the sort order when times and subject identifiers match.

The plus operator means "outer join" or "full merge" when the arguments are "keyed" data.frames. The pipe operator means "left join" (merge, all.x=TRUE) when the arguments are "keyed" data.frames.

Listing 15:

```
> dat <-
+   nm() +
+   aug(dose, SEQ=1, EVID=1) +
+   aug(pk,   SEQ=0, EVID=0) |
+   dem
```

```
outer join of 0 rows and 40 rows on SUBJ, SEQ, HOUR
outer join of 40 rows and 560 rows on SUBJ, SEQ, HOUR, EVID
left join of 600 rows and 40 rows on SUBJ
```

Listing 16:

```
> summary(dat)
```

	value
rows	600
records	600
comments	0
subjects	40
longestCase	72
naKeys	0
dupKeys	0
badDv	0
falseDv	0
zeroDv	25
predoseDv	40
badAmt	0
falseAmt	0
zeroAmt	0
noPk	0
badII	0

Note predose/zero DV. See ?zeroDv We comment-out these records.

Listing 17:

```
> dat <- hide(dat, where=predoseDv(dat), why='predose')
> summary(dat)
```

	value
rows	600
records	560
comments	40
subjects	40
longestCase	72
naKeys	0
dupKeys	0
badDv	0
falseDv	0
zeroDv	10

```
predoseDv      0
badAmt         0
falseAmt       0
zeroAmt        0
noPk           0
badII          0
```

We still have some zero DV that are not predose. We comment those as well.

Listing 18:

```
> dat <- hide(dat, where=zeroDv(dat), why='zerodv')
> summary(dat)
```

```
      value
rows      600
records   550
comments    50
subjects   40
longestCase 72
naKeys      0
dupKeys     0
badDv       0
falseDv     0
zeroDv      0
predoseDv   0
badAmt      0
falseAmt    0
zeroAmt     0
noPk        0
badII       0
```

Listing 19:

```
> head(dat)
```

```
  C SUBJ TIME SEQ HOUR EVID ID  AMT TAFD  TAD LDOS  DV MDV HEIGHT WEIGHT SEX
1 C    1 0.00  0 0.00   0  1   NA 0.00   NA   NA 0.000  0   174   74.2  0
2 .    1 0.00  1 0.00   1  1 1000 0.00 0.00 1000   NA   1   174   74.2  0
3 .    1 0.25  0 0.25   0  1   NA 0.25 0.25 1000 0.363  0   174   74.2  0
4 .    1 0.50  0 0.50   0  1   NA 0.50 0.50 1000 0.914  0   174   74.2  0
5 .    1 1.00  0 1.00   0  1   NA 1.00 1.00 1000 1.120  0   174   74.2  0
6 .    1 2.00  0 2.00   0  1   NA 2.00 2.00 1000 2.280  0   174   74.2  0
  AGE DOSE FED SMK DS CRCN predose zerodv
1 29.1 1000  1  0  0 83.5      1      1
2 29.1 1000  1  0  0 83.5      0      0
3 29.1 1000  1  0  0 83.5      0      0
4 29.1 1000  1  0  0 83.5      0      0
5 29.1 1000  1  0  0 83.5      0      0
6 29.1 1000  1  0  0 83.5      0      0
```

We could rearrange columns for convenience and clarity.

Listing 20:

```
> dat <- shuffle(dat,c('C','ID','TIME','SEQ','EVID','AMT','DV'))
> head(dat)
```

	C	ID	TIME	SEQ	EVID	AMT	DV	SUBJ	HOUR	TAFD	TAD	LDOS	MDV	HEIGHT	WEIGHT	SEX
1	C	1	0.00	0	0	NA	0.000	1	0.00	0.00	NA	NA	0	174	74.2	0
2	.	1	0.00	1	1	1000	NA	1	0.00	0.00	0.00	1000	1	174	74.2	0
3	.	1	0.25	0	0	NA	0.363	1	0.25	0.25	0.25	1000	0	174	74.2	0
4	.	1	0.50	0	0	NA	0.914	1	0.50	0.50	0.50	1000	0	174	74.2	0
5	.	1	1.00	0	0	NA	1.120	1	1.00	1.00	1.00	1000	0	174	74.2	0
6	.	1	2.00	0	0	NA	2.280	1	2.00	2.00	2.00	1000	0	174	74.2	0

	AGE	DOSE	FED	SMK	DS	CRCN	predose	zerodv
1	29.1	1000	1	0	0	83.5	1	1
2	29.1	1000	1	0	0	83.5	0	0
3	29.1	1000	1	0	0	83.5	0	0
4	29.1	1000	1	0	0	83.5	0	0
5	29.1	1000	1	0	0	83.5	0	0
6	29.1	1000	1	0	0	83.5	0	0

We create a file using write.nm to format NAs specially, etc.

Listing 21:

```
> write.nm(dat,file='../data/derived/phase1.csv')
```